

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Белгородский Валерий Савельевич

Должность: Ректор

Дата подписания: 08.06.2024 06:40:07

Уникальный программный ключ:

8df276ee93e17c18e7bee9e7ad240ee7ab824f3

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Российский государственный университет им. А.Н. Косыгина
(Технологии. Дизайн. Искусство)»

**МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ
ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ ПО
ДИСЦИПЛИНЕ
МДК 01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ**

специальность

09.02.07 Информационные системы и программирование

Москва 2024

В методических рекомендациях излагаются порядок выполнения курсового проекта, примеры разработки пунктов курсового проекта, содержание пояснительной записки, защита и порядок оценивания результатов работы.

Методические рекомендации предназначены для студентов второго курса специальности 09.02.07 «Информационные системы и программирование», изучающих МДК 01.01. «Разработка программных модулей».

Содержание

ВВЕДЕНИЕ	4
1.Содержание курсового проекта	6
2. Порядок выполнения курсового проекта.....	7
3.Содержание пояснительной записки.....	8
4. Примеры разработки пунктов курсового проекта.....	8
<i>Пункт 2.1 Функциональные требования к программному.....</i>	<i>8</i>
<i>Пункт 2.2 Структура пользовательского интерфейса Пункт</i>	
<i>2.2.1 Анализ полей класса</i>	<i>9</i>
<i>Пункт 2.2.2 Интерфейс пользователя</i>	<i>9</i>
<i>Пункт 2.2.3 Сценарий работы пользователя</i>	<i>10</i>
<i>Пункт 2.3 Архитектура программного модуля.....</i>	<i>11</i>
<i>Пункт 2.3.1 Архитектура формы «Жители г. Москвы».....</i>	<i>12</i>
ПРИМЕР (словесное описание).....	14
<i>Пункт 2.4 Описание алгоритма разработки модуля</i>	<i>15</i>
<i>Пункт 2.4.1Алгоритм метода «Видим ввод».....</i>	<i>15</i>
(ПРИМЕР –БЛОК-СХЕМА).....	15
<i>Пункт 2.5 Тестирование и отладка модуля.....</i>	<i>15</i>
<i>Пункт 2.6 Разработка документа «Руководство оператора» .</i>	<i>17</i>
<i>Пункт ПРИЛОЖЕНИЯ.....</i>	<i>17</i>
5. Защита курсового проекта	17
6. Примерная тематика курсовых проектов	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	22

ВВЕДЕНИЕ

Методические рекомендации предназначены для студентов второго курса специальности 09.02.07 «Информационные системы и программирование».

Выполнение курсового проекта студентами программы подготовки специалистов среднего звена (ППССЗ) специальности 09.02.07 «Информационные системы и программирование» рассматривается как вид промежуточной аттестации по МДК 01.01 «Разработка программных модулей» и реализуется в пределах времени, отведенного на изучение.

Целью данного курсового проекта является обобщение знаний, умений в области технологий структурного и объектно-ориентированного программирования, полученных студентами за время изучения МДК 01.01. «Разработка программных модулей» для овладения видом профессиональной деятельности «Разработка модулей программного обеспечения для компьютерных систем», формирование профессиональной компетентности обучаемых к практической деятельности в области разработки приложений, работающих под управлением современных операционных систем, с применением современных технологий и сред разработки программного обеспечения. Овладение следующими профессиональными компетенциями: формировать алгоритмы разработки программных модулей в соответствии с техническим заданием; разрабатывать программные модули в соответствии с техническим заданием; выполнять отладку программных модулей с использованием специализированных программных средств; выполнять тестирование программных модулей; осуществлять рефакторинг и оптимизацию программного кода. Овладение умением составления пояснительной записки, содержащей обоснование принятых проектных решений; применения нормативных документов, регламентирующих состав, содержание и форму документации на разработанный программный продукт.

В настоящее время клиент-серверная разработка — одна из самых востребованных отраслей программирования. Современная технология взаимодействия в информационных системах предполагает реализацию различных интерфейсов. Графический

интерфейс – это наиболее часто применяемый механизм, поддерживающий процесс взаимодействия пользователя с информационными ресурсами. Язык программирования C# — один из самых востребованных и динамично развивающихся языков программирования в отрасли информационных технологий, ориентированный на программирование в распределенной гетерогенной вычислительной среде и поддерживающий технологию объектно-ориентированного программирования.

Актуальность курсового проекта заключается в том, что дает возможность получить знания и умения практической разработки клиент-серверных приложений в технологии DOT.NET, с применением современных паттернов и инструментов программирования, с целью разработки многофункциональных информационных систем.

Изучение языка программирования C# позволяет приобрести фундаментальные знания, которые позволят освоить любые аспекты современного программирования. Выбор интегрированной среды Microsoft Visual Studio обусловлен тем, что это самый серьезный продукт производства Microsoft для профессиональной разработки программного обеспечения. Реализация именно программного модуля Windows Forms позволяет наиболее полно изучить инструмент, который дает возможность создавать графические интерфейсы и поддерживать механизм событийности, необходимый для взаимодействия пользователя с системой.

В соответствии с поставленной целью необходимо разработать программный модуль, обеспечивающий одну из реализаций на стороне клиента графического пользовательского интерфейса клиент-серверного приложения в среде Microsoft Visual Studio на языке программирования C# и для этого решить следующие задачи:

- разработать документ «Техническое задание на разработку программного модуля»;
- определить задачи и определить методы для функционирования приложения;
- обеспечить ввод необходимой информации; для этого реализовать методы, позволяющие организовать ввод исходных данных с экрана монитора; осуществить запись данных в файл;
- обеспечить возможность корректировки информации; для этого реализовать методы, позволяющие организовать корректировку информации по запросу пользователя;

- обеспечить вывод информации из файла на экран монитора; для этого реализовать методы, позволяющие осуществить вывод информации из файла ;
- обеспечить реализацию сортировки и фильтрации по разным полям;
- обеспечить реализацию запросов для поиска по заданным критериям; для этого реализовать методы, позволяющие осуществить поиск по заданным критериям с выводом полученных результатов на экран монитора и в файл ;
- разработать тест - требования и тестовые примеры для проведения тестирования модуля;
- провести тестирование по разработанным требованиям и подготовить отчет по проблемам;
- разработать документ «Руководство оператора».

1.Содержание курсового проекта

Выполнение курсового проекта разбивается на два этапа.

Первая глава (теоретическая часть) посвящается рассмотрению основополагающих теоретических вопросов, которые должны охарактеризовать проблемную область и раскрыть тематику исследования. Приводимые понятия и определения должны раскрывать и дополнять последовательность рассуждений студента о проблемной области в рамках темы курсового проекта, формируя логическую последовательность изложения, подводящего непосредственно к рассматриваемому объекту исследования.

Эти теоретические аспекты должны послужить основой для раскрытия темы курсового проекта. Эта часть курсового проекта не самоцель, а всего лишь средство для более полного раскрытия и всестороннего освещения избранной темы. Исследование теоретических вопросов, содержащихся в первой главе, должно быть увязано с практической частью работы и служить базой для разработки практической части.

Во второй главе (практическая часть) на основе существующих практических подходов и, опираясь на теоретические положения, которые были освещены автором в первой главе, рассматривается, анализируется и реализуется фактический материал по разрабатываемому вопросу, в плане применения к конкретной проблемной области (разработка архитектуры программного модуля,

разработка интерфейса, разработка алгоритмов реализации модуля, разработка планов тестирования, разработка документации по поддержке программного модуля).

Заключение суммирует итоги курсового проекта, отражает основные результаты, достигнутые при решении вопросов и проблем, поставленных в исследуемой теме. Заключение должно быть кратким и вместе с тем емким, обобщая, систематизируя и углубляя выводы, сделанные по итогам всего курсового проекта. Оно должно служить подтверждением решения задач, поставленных во введении, включая теоретические и практические аспекты.

2. Порядок выполнения курсового проекта

В процессе выполнения курсового проекта студенты должны:

1. изучить задание на выполнение курсового проекта;
2. разработать теоретическую часть курсового проекта: ГЛАВА 1. ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ;
3. выполнить анализ задания на выполнение курсового проекта и, руководствуясь пунктами пп.4.-12 порядка выполнения курсового проекта, разработать практическую часть курсового проекта: ГЛАВА 2. РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ПРИЛОЖЕНИЯ;
4. подготовить документ «Техническое задание на разработку программного модуля»;
5. провести анализ документа «Техническое задание на разработку программного модуля» в части функциональные требования к программному модулю;
6. выбрать структуры данных для реализации предметной области программного модуля;
7. разработать интерфейс пользователя;
8. выбрать стратегию тестирования и разработать тесты;
9. выбрать и обосновать выбор языка и среды программирования;
10. разработать алгоритмы и реализовать их в коде в выбранной среде разработки;
11. выполнить тестирование и отладку;
12. разработать необходимую документацию: документ «Руководство оператора»

3.Содержание пояснительной записки

Содержание пояснительной записки обязательно должно состоять из следующих пунктов:

ВВЕДЕНИЕ

ГЛАВА1.ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ

1.1. Структура типичного приложения.

1.2. Реализация графического интерфейса в .NET.

1.3. Требования к дизайну пользовательского интерфейса.

ГЛАВА 2. РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО

ИНТЕРФЕЙСА ПРИЛОЖЕНИЯ

2.1.Функциональные требования к программному модулю.

2.2. Структура пользовательского интерфейса .

2.2.1.Анализ полей класса

2.2.2. Интерфейс пользователя

2.2.3. Сценарий работы пользователя

2.3.Архитектура программного модуля.

2.3.1 Архитектура формы «имя формы»

2.3.2 и т.д.

2.4 Описание алгоритма разработки модуля

2.4.1 Алгоритм метода «имя метода»

2.4.2 и т.д.

2.5.Тестирование и отладка модуля

2.6 Разработка документа «Руководство оператора»

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

ПРИЛОЖЕНИЕ

4. Примеры разработки пунктов курсового проекта

Пункт 2.1 Функциональные требования к программному модулю

Программный модуль должен обеспечивать одну из реализаций на стороне клиента графического пользовательского интерфейса клиент-серверного приложения **ваша тема.....**

Для функционирования приложения программный модуль должен: **(как минимум)**

- обеспечить ввод необходимой информации, организовать ввод исходных данных с экрана монитора;

- осуществить запись данных в файл;
- обеспечить возможность корректировки информации по запросу пользователя;
- обеспечить вывод информации из файла на экран монитора;
- обеспечить реализацию сортировки и фильтрации по разным полям;
- обеспечить реализацию запросов для поиска по заданным критериям с выводом полученных результатов на экран монитора.

Пункт 2.2 Структура пользовательского интерфейса

Пункт 2.2.1 Анализ полей класса

На основании анализа функциональных требований к разрабатываемому программному модулю проведен анализ полей класса и результат представлен в таблице 1.

Таблица 1. Анализ полей класса

<i>Наименование</i>	<i>Идентификатор</i>	<i>Тип данных</i>	<i>Размерность (максимум)</i>	<i>Допустимые значения</i>
Фамилия	Surname	строковый string	35 символов кириллицы	От А до Я
Имя	Name	строковый string	35 символов кириллицы	От А до Я
Отчество	Patronymic	строковый string	35 символов кириллицы	От А до Я
Серия паспорта	seripasport	числовой integer	4 символа цифровых	От 0 до 9

Пункт 2.2.2 Интерфейс пользователя

Интерфейс пользователя (user interface или сокращенно UI) – это интерфейс, с помощью которого человек может управлять программным обеспечением или аппаратным оснащением.

Программный модуль представляет собой графический интерфейс, реализованный с помощью следующих форм: (Пример)

- Главная форма «Жители г. Екатеринбурга» Form1.cs
- Форма «Авторизации» – AuthorizationForm.cs
- Форма «Регистрации» – RegistrationForm.cs
- Форма «О программе AboutTheProgramForm.cs

Интерфейс представлен на рис. 1. Все надписи – это объекты *Label*. Поля ввода – это объекты *TextBox*. Для отображения списка найденных записей используется объект *comboBox1*. Надпись на кнопке «*BUTTON3*» изначально не сформирована. Надпись появится после выбора варианта работы: при вводе данных – надпись «Сохранить», при поиске – «Поиск». Порядок появления объектов описан в сценарии.

The screenshot displays a graphical user interface for a program. At the top, there are two buttons: "ВВОД ДАННЫХ" (Data Entry) and "ПОИСК ДАННЫХ" (Search Data). Below these are several input fields and a dropdown menu. The fields are labeled: "ФИО" (Full Name), "Пол" (Gender), "Дата рождения" (Date of Birth), "Улица" (Street), "Дом" (House), "Квартира" (Apartment), and "Общая площадь" (Total Area). A "ВЫБРАТЬ" (Select) button is positioned next to the "Пол" field. At the bottom of the form, there are three buttons: "button3", "ОТМЕНИТЬ" (Cancel), and "ЗАКРЫТЬ" (Close).

Рис.1.Интерфейс пользователя (главная форма)

Пункт 2.2.3 Сценарий работы пользователя

Изначально на экране отобразим только две кнопки: «**ВВОД ДАННЫХ**» и «**ПОИСК ДАННЫХ**». Рассмотрим оба случая.

1. При нажатии на кнопку **«ВВОД ДАННЫХ»** появляются поля ввода данных для набора и сохранения новых данных. Вместе с полями появляются кнопки **«СОХРАНИТЬ»**, **«ОТМЕНИТЬ»** и **«ВЫБРАТЬ»**. Кнопка **«ВЫБРАТЬ»** обеспечивает выбор пола: М или Ж. Должны быть заполнены все поля ввода. При нажатии кнопки **«СОХРАНИТЬ»** проверяется правильность заполнения полей (все ли заполнено) и данные сохраняются в файле – выполняется запись в конец файла. Пользователь информируется о записи сообщением на экран: «Сведения добавлены». При нажатии кнопки **«ОТМЕНИТЬ»** ничего не происходит, поля ввода скрываются, и вновь на экране отображаются две исходные кнопки.

2. При нажатии на кнопку **«ПОИСК ДАННЫХ»** отображаются все поля для набора данных, кроме номера квартиры и площади жилья. Отображаются также кнопки **«ПОИСК»**, **«ВЫБРАТЬ»** и **«ОТМЕНИТЬ»**. В поля пользователь может набрать значения – в каждое поле или только в некоторые. Введенные значения являются ключами поиска информации. При нажатии кнопки **«ОТМЕНИТЬ»** ничего не происходит, поля ввода скрываются, и вновь на экране две исходные кнопки. При нажатии на кнопку **«ПОИСК»** выполняется поиск записей в файле. После завершения поиска на экране отображается список найденных данных в виде ФИО. Список может быть и пустым, если при поиске не найдено ни одной записи. Вместе со списком отображается кнопка **«ЗАКРЫТЬ»**. Теперь пользователь может выбрать любой элемент списка, и на экране отображается полная информация.

При нажатии на кнопку **«ЗАКРЫТЬ»** поля ввода скрываются, и вновь на экране – две исходные кнопки.

Пункт 2.3 Архитектура программного модуля

Когда речь идет о построении архитектуры программного модуля, создании его структуры, под этим, главным образом, подразумевается декомпозиция программного модуля на подсистемы (функциональные методы) и организация их взаимодействия друг с другом и внешним миром. Причем, чем более независимы методы, тем безопаснее сосредоточиться на разработке каждого из них в отдельности в конкретный момент времени и при этом не заботиться обо всех остальных частях.

В этом случае программа превращается в конструктор, состоящий из набора подпрограмм, взаимодействующих друг с другом по хорошо определенным и простым правилам, что собственно и позволяет контролировать ее сложность, а также дает возможность получить все те преимущества, которые обычно соотносятся с понятием хорошая архитектура.

"Архитектура идентифицирует главные компоненты системы и способы их взаимодействия. Также это выбор таких решений, которые интерпретируются как основополагающие и не подлежащие изменению в будущем".

"Архитектура — это организация системы, воплощенная в ее компонентах, их отношениях между собой и с окружением.

Система — это набор компонентов, объединенных для выполнения определенной функции".

В курсовом проекте разработки модуля в технологии Windows Forms архитектура программного модуля должна быть представлена в виде:

- таблиц, описывающих управляющие элементы каждой из форм;
- структурных схем реализации форм.

Структурная схема отражает состав и взаимодействие по управлению частей разрабатываемого программного модуля. В курсовом проекте структурные компоненты - это методы, реализующие архитектуру программного модуля.

Следовательно, для каждой из форм необходимо представить структурную схему реализации.

Пункт 2.3.1 Архитектура формы «Жители г.Екатеринбурга»

Таблица 2. Управляющие элементы формы «Жители г.Екатеринбурга»

Номер управляющего элемента	Имя элемента	Какие действия выполняются
Button1	Ввод данных	Открывает поля для ввода данных
Button2	Поиск данных	Открывает поля для

		набора данных. Отображает кнопки «Поиск» «Выбрать» «Отменить»
И.т.д.		

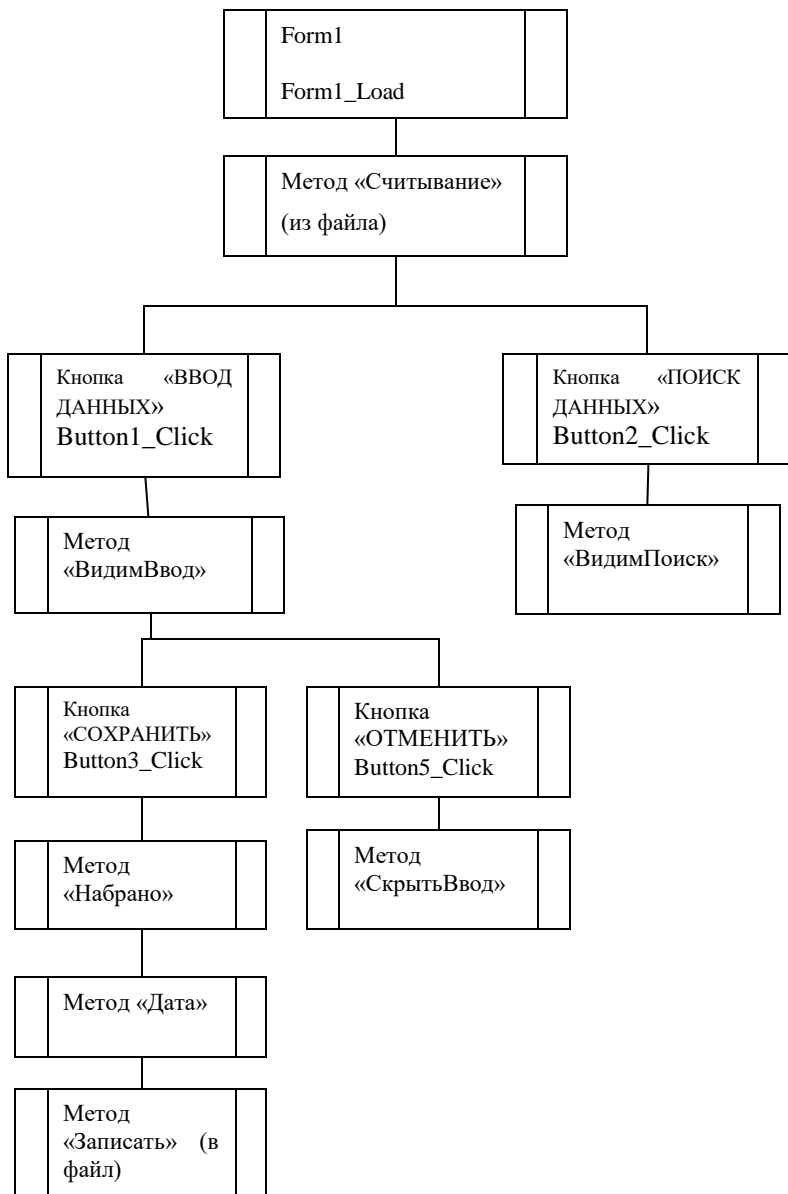


Рис.2 Структурная схема реализации формы «Жители г.Екатеринбурга». (ПРИМЕР)

ПРИМЕР (словесное описание).

Реализация ввода данных предусматривает разработку следующих методов:

«Видим ввод»

«Набрано»

«Сохранить»

И т.д.

Пункт 2.4 Описание алгоритма разработки модуля

В данном пункте на основе структурных схем реализации необходимо представить алгоритм каждого из методов в виде словесного описания или блок-схемы.

Пункт 2.4.1 Алгоритм метода «Видим ввод» **(ПРИМЕР словесное описание)**

1. Установить видимость всех надписей, полей ввода и кнопок.
2. Установить пустые значения для всех полей.
3. И т.д.

(ПРИМЕР –БЛОК-СХЕМА)



Рис3. Алгоритм метода «Видим ввод»

Пункт 2.5 Тестирование и отладка модуля

В данном пункте: на основе контрольного списка разработать тест-требования для тестирования пользовательского интерфейса, разрабатываемого в вашем курсовом проекте; провести тестирование пользовательского интерфейса, составить отчет о проблеме в виде таблицы.

Таблица 3. Отчет о проблеме (**ПРИМЕР**)

№ п/п	Тест-требования	Результат тестирования (кратко-да\нет; выполняется \не выполняется и т.д. по смыслу требования)
<i>кнопки</i>		
1	Все кнопки, запускающие действия, имеют текст в инфинитивной форме глагола (пример: искать), а не другую часть речи, либо форму глагола (пример: готово). Давать кнопке текст «ОК» можно, только если какой-либо глагол не вмещается.	Выполняется
2	Между кнопками, стоящими рядом, должно быть пустое пространство, щелчок по которому не обрабатывается.	Выполняется
<i>Поля ввода</i>		
3	В полях ввода уже стоят наиболее вероятные значения (значения по умолчанию).	Выполняется
4	Длина полей не меньше, и, по возможности, не больше, длины вводимых в них данных.	Выполняется

	И т.д.	

Пункт 2.6 Разработка документа «Руководство оператора»

Ознакомиться с ГОСТ 19.505-79 Межгосударственный стандарт. Единая система программной документации. Руководство оператора.

В соответствии с данным стандартом разработать и оформить документ «Руководство оператора».

Пункт ПРИЛОЖЕНИЯ

Ознакомиться с ГОСТ 19.201-78 Единая система программной документации. Техническое задание. Требования к содержанию и оформлению.

Оформить документ «Техническое задание» на разработку программного модуля.

5. Защита курсового проекта

Защита курсового проекта осуществляется в установленные сроки. К защите допускаются курсовые проекты, выполненные в установленные сроки и имеющие положительный отзыв руководителя о ходе проектирования. Решение о возможности допуска проекта к защите принимает руководитель проекта.

Защита (публичная защита) проводится в присутствии представителей преподавательского состава колледжа (заседание комиссии).

Защита включает в себя доклад студента на основе компьютерной презентации и демонстрацию разработанного программного продукта.

В выступлении следует сформулировать цели и задачи курсового проекта, раскрыть его структуру, показать используемые при проектировании решения (методы, способы и средства решения

отдельных программных задач). Следует уделить внимание выводам, сделанным автором на основе проделанной работы. Длительность выступления составляет 5-7 минут.

После доклада члены комиссии задают студенту вопросы, связанные с проектированием. Далее студент демонстрирует работу разработанного программного продукта.

По окончании защиты члены комиссии на закрытом заседании коллективно обсуждают итоги защиты каждого проекта и оценивают большинством голосов по 5-ти бальной системе. При равном количестве голосов приоритетное право решения предоставляется руководителю проекта..

В тех случаях, когда защита курсового проекта признается неудовлетворительной, студент может представить к повторной защите тот же проект с доработкой, определяемой комиссией, или же обязан разработать новую тему, которая устанавливается кафедрой. Сроки и условия защиты проекта в этом случае устанавливаются кафедрой по согласованию с заместителем директора по учебной работе.

Критериями, влияющими на результирующую оценку являются:

- 1) полнота реализации требований к программному модулю;
- 2) удобство пользовательского интерфейса;
- 3) стиль написания программного кода;
- 4) тщательность тестирования программных модулей;
- 5) качество оформления пояснительной записки;
- 6) полнота и правильность ответов на вопросы;
- 7) соблюдение календарного плана выполнения работы.

Также к разработанному программному продукту применяются критерии оценки программного продукта согласно показателям качества по ГОСТ 28195-89 (таблица 4):.

Таблица 4. Показатели качества программного продукта

№ п/п	Показатели качества
1	Показатели надежности программного продукта: - устойчивость функционирования

	- работоспособность
2	Показатели сопровождения: - структурность - простота конструкции - наглядность - повторяемость
3	Показатели удобства применения - легкость освоения - доступность эксплуатационных программных документов - удобство эксплуатации и обслуживания

Оценка «отлично» выставляется за работу, выполненную в установленные сроки, полностью отвечающую критериям оценки 1)-б) и созданный программный продукт разработан в полном соответствии с показателями качества .

Оценка «хорошо» выставляется за работу, выполненную в установленные сроки, в случае, если программный продукт удовлетворяет критериям оценки 1) и не удовлетворяет некоторым критериям 2)-б), и созданный программный продукт имеет несоответствие одному из показателей качества.

Оценка «удовлетворительно» выставляется за работу, выполненную в установленные сроки, если программный продукт не полностью удовлетворяет критерию оценки 1) и по сумме всех требований 2)-б) уточняется в зависимости от качества работы, и созданный программный продукт имеет несоответствие по двум показателям качества .

Оценка «неудовлетворительно» выставляется в том случае, если сроки выполнения этапов не соблюдены, работа выполнена частично особенно в части полноты реализации требований к программному модулю (критерий 1)) и по сумме всех требований 2)-б) уточняется в зависимости от качества работы, и созданный программный продукт не соответствует более двум показателям качества.

Также оценка «неудовлетворительно» выставляется в случае, если в ходе защиты установлено, что студент не владеет материалом.

6. Примерная тематика курсовых проектов

1. Разработка программного модуля «Дорожно-транспортные происшествия».
2. Разработка программного модуля «Телефоны и абоненты».
3. Разработка программного модуля «Сведения о книжном фонде библиотеки».
4. Разработка программного модуля «Продажа авиабилетов».
5. Разработка программного модуля «Обувной магазин».
6. Разработка программного модуля Кафедра».
7. Разработка программного модуля «Интернет магазин».
8. Разработка программного модуля «Преподаватели».
9. Разработка программного модуля «Отдел кадров предприятия».
10. Разработка программного модуля «Отдел снабжения предприятия».
11. Разработка программного модуля «Ремонтная мастерская».
12. Разработка программного модуля «Магазин бытовой техники».
13. Разработка программного модуля «Аукционы».
14. Разработка программного модуля «Кинотеатры (Афиша)».
15. Разработка программного модуля «Кинотеатры (Размещение и сеансы)».
16. Разработка программного модуля «Ресторан».
17. Разработка программного модуля «Бюро знакомств .
18. Разработка программного модуля «Продажа жилья».
19. Разработка программного модуля «Клиент отеля ».
20. Разработка программного модуля «Нарушители правил дорожного движения».
21. Разработка программного модуля «Путевой лист для перевозки груза».
22. Разработка программного модуля «Расписание электричек».
23. Разработка программного модуля «Продажа автомобилей».
24. Разработка программного модуля «Туристические путевки».
25. Разработка программного модуля «Трудоустройство».
26. Разработка программного модуля «Выставка собак».
27. Разработка программного модуля «Научные труды сотрудников».
28. Разработка программного модуля «Пассажирское судоходство».
29. Разработка программного модуля «Приемные экзамены».
30. Разработка программного модуля «Ипподром».
31. Разработка программного модуля «Справочник филателиста».

32. Разработка программного модуля «Расписание занятий студента».
33. Разработка программного модуля «Пассажир поезда дальнего следования».
34. Разработка программного модуля «Цех предприятия».
35. Разработка программного модуля «Читатели студенческой библиотеки».
36. Разработка программного модуля «Записная книжка».
37. Разработка программного модуля «Телепрограмма».
38. Разработка программного модуля «Промышленное рыболовство».
39. Разработка программного модуля «Страховые иски».
40. Разработка программного модуля «Учет успеваемости в колледже».
41. Разработка программного модуля «Автобусные маршруты».
42. Разработка программного модуля «Обработка заказов».
43. Разработка программного модуля «Спортивные рекорды».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васюткина, И.А. Разработка клиент-серверных приложений на языке C#: учебное пособие: Новосибирск : Новосибирский государственный технический университет, 2020. – 112 с. : ил., табл. – Режим доступа: по подписке. [<http://biblioclub.ru/index.php?page=book&id=576570> 10.04.23].
2. Горелов, С.В. Современные технологии программирования: разработка Windows-приложений на языке C#: Финансовый университет при Правительстве Российской Федерации. – М.: Прометей, 2019. – 363 с.: ил. – Режим доступа: по подписке. [<http://biblioclub.ru/index.php?page=book&id=576037> 10.04.2023]
3. Интегрированная среда разработки Microsoft Visual Studio. Режим доступа: [<https://visualstudio.microsoft.com> 20.04.2023].
4. ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению.– Введ. 1980. – 01.01. – М.: Изд-во стандартов СССР, 1978. – 10с. Режим доступа [<http://docs.cntd.ru/document/1200007648> 26.05.2023].
5. ГОСТ 19.701-90 ЕСПД Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.[электронный ресурс] // Консультант Плюс. ВерсияПроф. Режим доступа [<http://docs.cntd.ru/document/9041994> 26.05.2023].
6. ISO/IEC 6592:2000 Информационные технологии. Руководство по разработке документации для автоматизированных прикладных систем. Режим доступа [<https://metrology.com.ua/ntd/skachat-iso-iec-ohsas/iso/iso-iec-6592-2000/> 27.05.2023].
7. ISO/IEC 18019:2004 Разработка программного обеспечения и проектирование систем. Руководство по разработке и подготовке документации для пользователя по применению программного обеспечения. Режим доступа: [<https://www.hse.ru/data/2018/11/28/1144395257/ГОСТы%20и%20ЕСПД%202018-2019.pdf> 25.05.2022].
8. Макки А. Введение в .NET 4.0 и Visual Studio 2015 для профессионалов / А. Макки. - М.: Вильямс, 2022. - 350 с.
9. Постолиит Анатолий. Visual Studio .NET: разработка приложений баз данных / Анатолий Постолиит. - М.: БХВ-Петербург, 2021. - 544 с.

10. Джозеф Албахари. С# 5.0. Справочник. Полное описание языка; Диалектика / Вильямс - М., 2019. - 545 с.
11. Васильев Алексей С#. Объектно-ориентированное программирование; Питер - М., 2019. - 320 с.
12. Разработка интерфейса пользователя программного обеспечения. Режим доступа: [<https://coderlessons.com/tutorials/akademicheskii/programmnaia-inzheneriia/razrabotka-interfeisa-polzovatelja-programmnogo-obespecheniia> 24.05.2023].
13. Федеральный центр информационно-образовательных услуг. Режим доступа: [<http://fcior.edu.ru/>] 14.05.2023
14. Федеральные образовательные ресурсы. Режим доступа: [<http://www.edu.ru/>] 14.05.2021
15. Библиотека учебных курсов Microsoft [Электронный ресурс].- Режим доступа: [<http://msdn.microsoft.com/ru-ru/gg638594>], 14.05.2023
16. Единая система программной документации [Электронный ресурс]. - Режим доступа: <http://prog-cpp.ru/espd/>, 18.05.2023
17. 3